```
  REM Sort by the Average grade field in the range in descending order
  oSortFields(0).Field = 7
  oSortFields(0).SortAscending = FALSE

  REM Set the sort fields to use
  oSortDesc(0).Name = "SortFields"
  oSortDesc(0).Value = oSortFields()

  REM Now sort the range!
  oRange.Sort(oSortDesc())
End Sub
```

	A	B	C	D	E	F	G	H
1	**Student**	**HW #1**	**HW #2**	**HW #3**	**Quiz #1**	**Quiz #2**	**Test #1**	**Average**
2	**Ian**	100	100	91	90	100	96	96.17
3	**Emily**	100	100	81	100	75	94	91.67
4	**Andrew**	90	100	82	90	88	92	90.33
5	**Bethany**	95	100	82	80	88	93	89.67
6	**Jennifer**	85	93	73	80	100	90	86.83
7	**Charles**	80	93	73	80	75	84	80.83
8	**Haley**	85	93	82	70	75	76	80.17
9	**David**	75	86	91	40	88	79	76.50
10	**Ferdinand**	85	93	73	60	50	72	72.17
11	**Georgia**	70	80	55	39	75	67	64.33
12								

Figure 462: Grading sheet after sorting by average grade in descending order

Sorting a table using multiple columns

As with the Sort dialog, a range can be sorted using up to three columns or rows in a macro. Sorting with extra columns or rows is as easy as adding extra sort fields to a sort descriptor. The macro in Listing 22 again uses the grade sheet example from Figure 452 to illustrate how to sort by two columns. Figure 463 shows the results of this operation – note that records are sorted in ascending order, first by Quiz #1 scores and then by Quiz #2 scores.

Listing 22: SortByQuizScores sorts the grade sheet data range (A1:H11) using two columns

```
Sub SortByQuizScores
  Dim oSheet
  Dim oRange
  Dim oSortFields(1) as new com.sun.star.util.SortField
  Dim oSortDesc(0) as new com.sun.star.beans.PropertyValue

  oSheet = ThisComponent.Sheets(0)

  REM Set the range on which to sort
  oRange = oSheet.getCellRangeByName("A1:H11")
```